

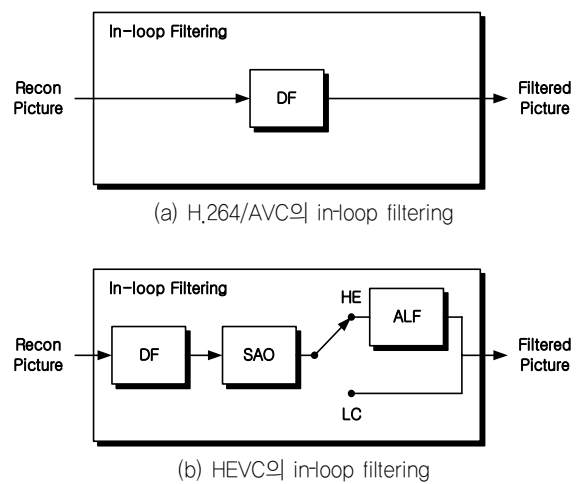
In-loop filtering in HEVC

양정엽 · 원광현 · 전병우 (성균관대학교)

I. 서론

최근의 멀티미디어 서비스는 HD(High Definition) 및 UHD(Ultra High Definition) 등과 같은 고해상도/고화질 영상을 요구하는 환경으로 급속히 천이하고 있다. 그러나, 현재 가장 우수한 압축 효율을 가진 것으로 알려진 H.264/AVC 표준^[1]과 같은 최신 압축 표준으로도 이러한 방대한 데이터를 저장하거나 현재의 통신채널로 손쉽게 전송하기에는 상당한 비용이 발생하고 있다. 이러한 문제에 대응하고자, 영상 압축 표준의 대표적인 양대 표준화 기구인 ISO/IEC JTC1/SC29/WG11 MPEG(Moving Picture Experts Group)과 ITU-T Q6/WP3/SG16 VCEG(Video Coding Experts Group)은, 지난 2010년 4월 독일 Dresden에서 개최된 제1회 JCT-VC (Joint Collaborative Team on Video Coding) 표준화 회의를 시작으로 차세대 영상 압축 표준인 HEVC(High Efficiency Video Coding) 표준 제정 프로젝트를 시작하였다^[2]. 현재 표준화가 진행되고 있는 HEVC 표준은 핵심원리에서는 기존의 H.264/AVC 표준과 같은 전통적인 움직임보상 변환부호화 압축 기술에 기반하고 있으나, 보다 향상된 다양한 요소 기술을 적용함으로써 압축 효율을 극대화하고 있다.

본 고에서 설명하고자 하는 루프내 필터링(In-loop filtering) 기술 역시 블록화 현상(Blocking artifact)을 제거하기 위한 디블록킹 필터(DF: Deblocking Filter)만을 포함하던 종래의 수동적인 루프내 필터링 기술과 달리, DF 뿐만 아니라, 양자화 등의 손실 압축에 의해 발생하는 정보의 손실을 보상하기 위한 SAO(Sample Adaptive Offset) 및 ALF(Adaptive Loop Filter)와 같은 새로운 요소기술을 추가적으로 포함함으로써, 주관적 화질 뿐만 아니라 압축 효율을 스스로 향상시킬 수 있는 매우 정교하면서도 능동적인 구조를 가지고 있다. 따라서



〈그림 1〉 H.264/AVC와 HEVC in-loop filtering의 블록도 비교

〈그림 1〉과 같이, 디블록킹 필터링만을 수행하는 H.264/AVC 표준의 인루프 필터링과 달리, HEVC 표준의 테스트 모델인 HM(HEVC Test Model)3.0^[3]에 포함된 인루프 필터링은 HE(High Efficiency) 조건인 경우에는 DF→SAO→ALF 순서로 수행되고, LC(Low Complexity) 조건인 경우에는 ALF가 제외되므로 DF→SAO 순서로 수행된다.

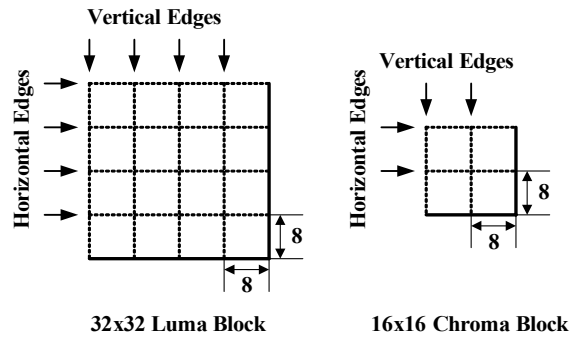
II. DF(Deblocking Filter) 기술

H.264/AVC 표준의 디블록킹 필터는 후처리 필터링(Post filtering) 과정을 통해 출력 영상의 주관적 화질만을 향상시키던 종래의 디블록킹 필터링 기술과는 달리, 복호화 과정 내에 디블록킹 필터링 과정을 포함시킴으로써 필터가 적용된 영상을 참조 영상으로 활용하는 루프내 필터링(In-loop filtering)

을 수행한다. 이러한 루프내 더블록킹 필터링 기술은 예측 및 양자화로 인해 발생하는 블록 경계의 왜곡을 효과적으로 제거함으로써 주관적 화질을 향상시키는 한편, 더블록킹 필터링이 적용된 영상을 참조 영상으로 활용함으로써 약 0.5dB 내외의 부호화 효율을 달성할 수 있는 유용한 기술로 알려져 있다^[4]. 특히, H.264/AVC 표준은 블록 경계에서 발생하는 왜곡의 정도에 따라 차별적인 필터링을 적용하는 적응적(Adaptive) 더블록킹 필터링 기술을 채택하고 있다. 즉, 블록 경계에서, 왜곡이 발생할 가능성이 높은 영역과 그렇지 않은 영역을 구분해 서로 다른 필터링을 수행함으로써 불필요한(또는 과도한) 필터링에 의해 발생할 수 있는 새로운 왜곡을 최소화한다. 그러나 이러한 적응적 더블록킹 필터링 기술은 왜곡의 정도를 측정하고 서로 다른 필터를 적용하기 위해 상당한 연산량을 요구하는 과정이기도 하다. 현재 표준화를 진행 중인 HEVC 표준의 HM3.0에 포함된 더블록킹 필터링 기술은, 전술한 적응적 인루프 더블록킹 필터링 기술의 개념을 상당한 정도 유사하게 채용하지만 동시에, 복잡한 연산 과정을 간략화시킨 방법으로서 Ericsson, Nokia 및 Tandberg가 공동으로 제안한 방법이 TMuC(Test Model under Consideration)^[5]에 포함된 이후 현재 HM3.0까지 유지되고 있다.

1. Filtering Unit(FU)

일반적으로 블록킹 현상은 예측 및 양자화에 의해 발생^[4]하기 때문에 PU(Prediction Unit) 및 TU(Transform Unit) 경계 모두에서 더블록킹 필터링이 수행되어야 한다. 종래의 H.264/AVC 표준은 PU 크기보다 TU 크기가 클 수 없기 때문에 TU 경계에서만 더블록킹 필터링을 수행하면 PU 경계를 포함하여 더블록킹 필터링을 자동적으로 수행할 수 있었다. 그러나 HM3.0에서는 화면간 예측 모드인 경우, PU 크기보다 TU 크기가 클 수 있으므로 TU 경계에 대해서만 더블록킹 필터링을 수행할 경우 PU 경계를 위한 더블록킹 필터링이 수행될 수 없는 경우가 발생한다. 따라서 HM3.0의 더블록킹 필터링은 PU 및 TU 경계 모두에서 수행된다. 또한, H.264/AVC 표준은 4x4 크기의 TU만을 사용하므로 수평 또는 수직 방향으로 4 화소 크기의 필터링 유닛만을 정의하고 있으나, HM3.0은 수평 또는 수직 방향으로 8 화소 크기의 필터링 유닛을 정의함으로써 더블록킹 필터링의 복잡도를 간략화 시켰다. 이는, 영상의 해상도가 증가하고, 예측의 성능이 향상되면서 잔여 신호의 발생량이 감소하기 때문에 블록킹 현상 역시 감소할 것이라는 일반적 사실에 근거한 것이며, 이런 이유로 8 화소 이하의 PU 또는 TU 경계에서는 더블록킹 필터링을 수행하지 않는다. <그림 2>는 32x32 CU(Coding Unit)에



<그림 2> HM3.0 더블록킹 필터의 필터링 유닛

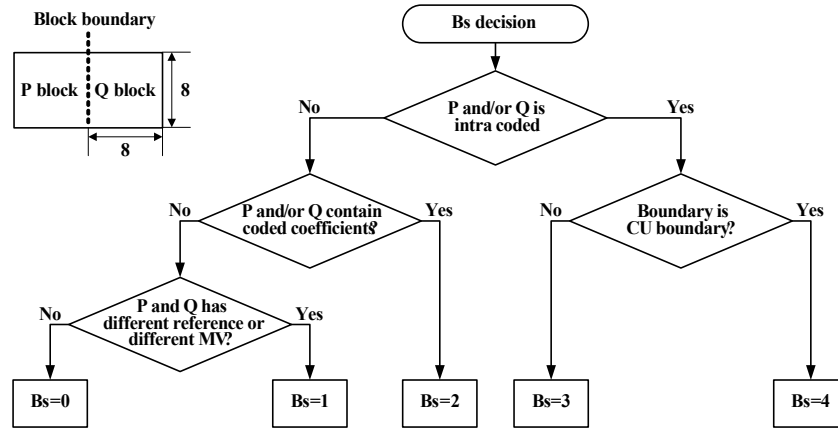
대한 필터링 유닛의 예를 보여준다. <그림 2>와 같이 밝기(Luma)값과 색차(Chroma)값 모두에 8 화소 단위의 필터링 유닛이 적용되며, 필터링 유닛 단위로 PU 및 TU 경계 여부에 따라 더블록킹 필터링 수행여부가 결정된다.

2. Bs(Boundary strength) 결정

필터링 유닛 단위로 더블록킹 필터링을 수행할 경계가 결정되면, 경계에 인접한 블록의 특성에 따라 적응적인 강도의 더블록킹 필터링을 수행하기 위해 경계세기 Bs(Boundary strength)를 결정한다. <그림 3>과 같이, HM3.0 더블록킹 필터링의 Bs 결정과정은 H.264/AVC 표준의 Bs 결정과정과 동일하다^[3,4]. 즉, 해당 경계가 CU 경계인가, 인접한 두 블록이 화면내 혹은 화면간 예측 모드로 부호화되었는가, 인접한 두 블록의 움직임 정보가 상이한가 등을 기준으로 0부터 4까지의 Bs를 결정하고, 결정된 Bs에 따라 더블록킹 필터링의 최대 강도를 조절함으로써 적응적인 더블록킹 필터링이 수행될 수 있게 한다. Bs가 0인 경우는 더블록킹 필터링이 수행되지 않음을 의미하고, Bs가 4인 경우는 가장 강한 더블록킹 필터링이 수행됨을 의미한다. Bs는 밝기값 블록을 기준으로 결정되고, 색차값 블록에는 밝기값 블록에 의해 결정된 Bs가 그대로 적용된다.

H.264/AVC 표준의 경우, Bs에 따라 더블록킹 필터링의 최대 강도를 조절하기 위해 각각의 Bs 별로 clipping value(t_c) table을 정의(기본적으로 clipping value는 QP에 의해 변하므로 QP에 따른 clipping value를 table로 정의)하고 있다^[4]. 반면, HEVC 경우는 간략화를 위해 <그림 4>와 같이, 하나의 clipping value(t_c) table만을 정의하는 대신, <식 1>과 같이 clipping value(t_c) offset을 0 또는 4로 정의하여 사용함으로써 더블록킹 필터링의 최대 강도를 조절한다^[3].

즉, 인접한 두 블록이 화면내 혹은 화면간 예측 모드로 부호화되었는가 여부에 따라 서로 다른 강도의 더블록킹 필터링이 적용됨을 의미한다.



〈그림 3〉 HM3.0에서의 Bs 결정과정

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
t _c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
QP	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
t _c	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4
QP	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	
t _c	5	5	6	6	7	8	9	9	10	10	11	11	12	12	13	13	14	14	

〈그림 4〉 Clipping value(t_c) table

$$\begin{aligned}
 t_c &= t_c(QP + 0) && \text{if } Bs = 1, 2 \\
 t_c &= t_c(QP + 4) && \text{if } Bs = 3, 4
 \end{aligned}
 \tag{식 1}$$

3. Filtering Decision

디블록킹 필터링을 수행할 경계 및 Bs가 결정되면, filtering decision 과정을 통해 필터링 유닛 단위로 필터링 수행 여부를 결정한다. 〈그림 5〉는 filtering decision 및 이후 설명할 filtering 과정을 설명하기 위해 수직 경계를 중심으로 인접한 P 및 Q 블록의 필터링 유닛 단위 화소를 보여준다. 수평

경계는 수직 경계와 동일한 방법으로 수행되므로 설명을 생략한다.

밝기값 블록의 경우, 〈식 2〉의 조건을 모두 만족하는 필터링 유닛에 대해 필터링이 수행되는 반면, 색차값 블록의 경우에는 Bs가 3 또는 4인 필터링 유닛에 대해 추가적인 filtering decision 과정없이 필터링이 수행된다.

〈그림 5〉 및 〈식 2〉에서 보는 바와 같이, d 값은 필터링 유닛에 포함된 8개의 화소줄(0번째 화소줄부터 7번째 화소줄까지)을 대표하는 2번째 및 5번째 화소줄을 이용해 계산되고, 〈그림 6〉과 같이 정의된 β 값과 비교된다. 즉, H.264/AVC

- Bs > 0
- $d = \sum |p_{2_i} - 2 * p_{1_i} + p_{0_i}| + \sum |q_{2_i} - 2 * q_{1_i} + q_{0_i}| < \beta$ where $i=2,5$ 〈식 2〉

	p3 ₀	p2 ₀	p1 ₀	p0 ₀		q0 ₀	q1 ₀	q2 ₀	q3 ₀	
	p3 ₁	p2 ₁	p1 ₁	p0 ₁		q0 ₁	q1 ₁	q2 ₁	q3 ₁	
	p3 ₂	p2 ₂	p1 ₂	p0 ₂		q0 ₂	q1 ₂	q2 ₂	q3 ₂	
Block P	p3 ₃	p2 ₃	p1 ₃	p0 ₃		q0 ₃	q1 ₃	q2 ₃	q3 ₃	Block Q
	p3 ₄	p2 ₄	p1 ₄	p0 ₄		q0 ₄	q1 ₄	q2 ₄	q3 ₄	
	p3 ₅	p2 ₅	p1 ₅	p0 ₅		q0 ₅	q1 ₅	q2 ₅	q3 ₅	
	p3 ₆	p2 ₆	p1 ₆	p0 ₆		q0 ₆	q1 ₆	q2 ₆	q3 ₆	
	p3 ₇	p2 ₇	p1 ₇	p0 ₇		q0 ₇	q1 ₇	q2 ₇	q3 ₇	

〈그림 5〉 P 및 Q 블록의 필터링 유닛 단위 화소 (수직 경계)

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7	8
QP	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
β	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32	34	36
QP	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	
β	38	40	42	44	46	48	50	52	54	56	58	60	62	64	64	64	64	64	

〈그림 6〉 β value table

표준은 filtering decision 과정을 위해 화소줄 단위의 연산을 수행하고 이를 α 및 β 값과 비교^[4]하는 반면, HEVC는 이러한 복잡한 과정을 필터링 유닛 단위로 수행^[3]함으로써 복잡도를 간략화 시키고 있음을 확인할 수 있다.

4. Filtering

전술한 모든 과정을 통해 필터링 강도 및 수행여부 등이 결정되면, 〈그림 5〉에서 보는 바와 같이, 필터링 유닛에 포함된 임의의 i 번째 화소줄 단위로 실제 더블록킹 필터링이 수행된다. 밝기값 블록을 위해서는 두 종류의 필터(weak 및 strong 필터)가 적용될 수 있다.

- $d < (\beta \gg 2)$
- $|p3_i - p0_i| + |q3_i - q0_i| < (\beta \gg 3)$ 〈식 3〉
- $|p0_i - q0_i| < ((5 * t_c + 1) \gg 1)$

〈식 3〉의 조건을 모두 만족하는 화소줄에 대해서는, 〈식 4〉

- $p0'_i = (p2_i + 2 * p1_i + 2 * p0_i + 2 * q0_i + q1_i + 4) \gg 3$
 - $q0'_i = (p1_i + 2 * p0_i + 2 * q0_i + 2 * q1_i + q2_i + 4) \gg 3$
 - $p1'_i = (p2_i + p1_i + p0_i + q0_i + 2) \gg 2$
 - $q1'_i = (p0_i + q0_i + q1_i + q2_i + 2) \gg 2$
 - $p2'_i = (2 * p3_i + 3 * p2_i + p1_i + p0_i + q0_i + 4) \gg 3$
 - $q2'_i = (p0_i + q0_i + q1_i + 3 * q2_i + 2 * q3_i + 4) \gg 3$
- 〈식 4〉

$$\Delta = \text{Clip}(-t_c, t_c, (13 * (q0_i - p0_i) + 4 * (q1_i - p1_i) - 5 * (q2_i - p2_i) + 16) \gg 5)$$

- $p0'_i = p0_i + \Delta$
 - $q0'_i = q0_i - \Delta$
 - $p1'_i = p1_i + \Delta/2$
 - $q1'_i = q1_i - \Delta/2$
- 〈식 5〉

$$\Delta = \text{Clip}(-t_c, t_c, (((q0_i - p0_i) \ll 2) + (q1_i - p1_i) + 4) \gg 3)$$

- $p0'_i = p0_i + \Delta$
 - $q0'_i = q0_i - \Delta$
- 〈식 6〉

와 같이 경계를 중심으로 3개의 화소를 필터링하는 strong 필터가 적용되고, 그렇지 않은 화소줄에 대해서는 〈식 5〉와 같이 경계를 중심으로 2개의 화소만을 필터링하는 weak 필터가 적용된다. 〈식 3〉에서 d 값은 전술한 filtering decision 과정에서 계산한 값을 그대로 사용하는 것이므로 필터링 유닛 단위의 연산이지만, 그 외의 모든 필터링 과정은 화소줄 단위로 수행된다.

Strong 필터는 H.264/AVC 표준에 정의된 special 필터^[4]와 동일하지만, weak 필터는 H.264/AVC 표준의 4-tap normal 필터^[4]와 달리 새롭게 정의된 6-tap 필터^[3]를 사용한다. 특히, weak 필터의 경우, H.264/AVC 표준에서는 경계를 중심으로 첫 번째 화소(p0_i 및 q0_i)와 두 번째 화소(p1_i 및 q1_i)를 위해서 서로 다른 4-tap 필터를 정의^[4]하고 있는 반면, HEVC는 필터링 과정의 복잡도를 감소시키기 위해 〈식 5〉와 같이 하나의 6-tap 필터를 사용하는 대신에 delta value를 조절하는 방법을 사용한다. 색차값 블록을 위해서는 하나의 필터만이 적용될 수 있으며, 전술한 바와 같이 경계에 인접한 P 및 Q 블록 중 하나 이상이 화면내 예측 모드로 부호화된

경우(Bs=3,4)에만 필터링을 수행한다. 필터링 과정은 <식 6>과 같이 경계를 중심으로 하나의 화소(p0, 및 q0)만이 필터링된다.

5. 관련 표준화 동향

현재까지 디블록킹 필터와 관련된 표준화 노력은 TE(Tool Experiments)-10, CE(Core Experiments)-8 및 CE-12를 거치면서 꾸준히 진행되고 있다. 특히, 현재 디블록킹 필터 관련 CE-12에서는 Ericsson, MediaTek, Microsoft, 성균관대학교 등의 기관을 중심으로 Bs 결정과정, filtering decision 및 filtering 과정 등의 개선을 통해 HM과 동일한 복잡도에서 평균 1% 이상의 부호화 효율 향상이 가능한 기술이 제안되고 있다. 뿐만 아니라, Sony, Panasonic 등의 기관을 중심으로는 디블록킹 필터링의 병렬 처리 등을 통한 복잡도 감소 노력도 함께 진행되고 있다.

III. SAO(Sample Adaptive Offset) 기술

SAO(Sample Adaptive Offset) 기술은 양자화 등의 부호화 과정을 통해 발생하는 원본 영상과 복원 영상 간의 왜곡을 화소(Sample) 단위의 offset을 통해 보상함으로써, 주관적 화질 및 부호화 효율을 같이 향상시키는 것이다. 즉, <식 7>과 같이, 부호화기가 N 개의 화소에 대해 원본 화소(org_n)와 복원 화소(rec_n) 간의 오차를 평균적으로 최소화하는 offset을 계산하고, 이를 복호화기에 전송하여 <식 8>과 같이, 부/복호화기가 동일한 방법으로 복원 화소의 왜곡을 보상할 수 있다. 특히, 왜곡의 정도가 다른 화소를 구분해 서로 다른 offset를 적용하는 적응적인(Adaptive) offset 보상 방법을 사용함으로써, 영상의 특성에 따라 부분적으로 다르게 발생하는 왜곡을 효율적으로 최소화할 수 있도록 한다.

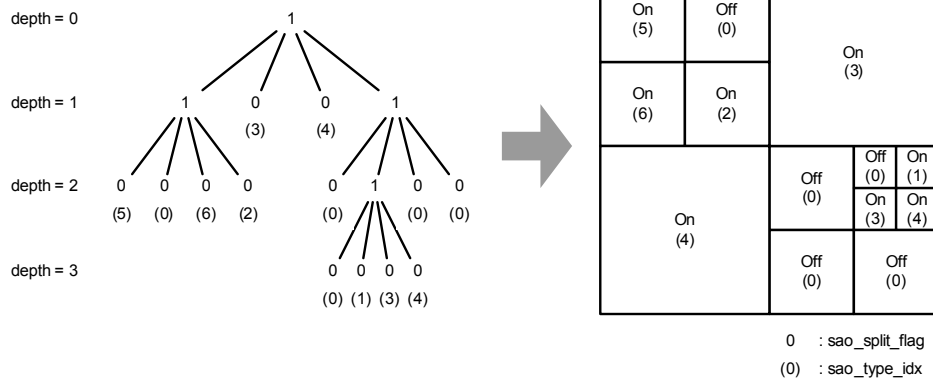
$$\text{offset} = \frac{\sum_{n=0}^{N-1} (\text{org}_n - \text{rec}_n)}{N} \quad \text{〈식 7〉}$$

$$\text{rec}'_n = \text{rec}_n + \text{offset} \quad \text{where } n=0,1, \dots, N-1 \quad \text{〈식 8〉}$$

디블록킹 필터와 달리, SAO 기술은 원본 영상과 복원 영상 간의 오류를 직접 계산하기 때문에 주관적 화질 뿐만 아니라 객관적 화질(PSNR)의 향상을 보장할 수 있다. 또한 SAO 역시 루프내 필터링 과정의 일부로 적용되어 왜곡이 보상된 영상을 참조 영상으로 활용함으로써 부호화 효율을 향상시킬 수 있다. 원본 영상과 복원 영상 간의 오류를 직접 계산하여 보상한다는 개념은 ALF(Adaptive Loop Filter)와 매우 유사하다. 특히 LC(Low Complexity) 조건에서는 ALF를 사용하지 않으므로, SAO가 ALF의 역할을 대체하는 간략화 된 기술이라 간주해도 무방할 것으로 보인다. SAO 기술은 지난 2010년 4월 독일 Dresden에서 개최된 제1회 JCT-VC 표준화 회의에서 삼성전자에 의해 BDC(BanD Correction) 및 EXC(EXTreme Correction) 등의 형태로 최초 제안^[6]된 이후, TE-10, CE-8, CE-13 등을 통해 MediaTek 제안서에 따라 현재의 구조로 개선되었다^[7-9]. 결국, 지난 2011년 4월 스위스 Geneva에서 개최된 제5회 JCT-VC 표준화 회의에서 평균 1% 이상의 부호화 효율(HE 조건에서 약 1.3%, LC 조건에서 약 1.8%)을 달성할 수 있음을 보여, HM3.0에 새롭게 채택된 기술이다^[10].

1. SAO Block Partitioning

일반적으로 양자화 과정을 통해 발생하는 왜곡은 예측의 정확도에 따라 서로 다르게 나타난다. 예를 들어, 정확한 예측이 수행된 블록은 양자화 과정에 의해 발생하는 왜곡이 거의 발생하지 않는 반면, 부정확한 예측이 수행된 블록은 잔여 신호가 많이 발생하므로 양자화 과정에서 왜곡이 많이 발생



<그림 7> SAO partitioning 구조 및 SAO type 적용의 예

할 가능성이 높다. 따라서 HM3.0의 SAO 기술은 왜곡의 정도에 따라 블록을 분할하고, 다양한 SAO 방법(SAO Type)을 적용할 수 있는 블록 단위의 적응적인 구조를 채택하고 있다^[10]. 즉, <그림 7>에서와 같이, quadtree 기반의 CU 분할 방법과 동일한 방식으로 LCU(Largest CU) 블록을 최대 4-depth까지의 하위 블록으로 분할할 수 있으며, 분할된 각 하위 블록에는 서로 다른 SAO type이 적용될 수 있다. 부호화기는 울-왜곡 최적화 방법을 사용하여 부호화 효율 측면에서 최적인 SAO 블록분할 여부 및 SAO type을 결정하고, 이에 해당하는 정보를 복호화기에 전송한다. 즉, SAO를 위한 블록분할 여부는 ALF와 달리, CU 블록분할 여부와 관계없이 독립적으로 결정될 수 있음을 의미한다. 따라서 <그림 7>의 예에서와 같이, 블록의 분할 여부를 표현하는 정보(숫자로 표시된 saosplitflag) 및 분할된 각 하위 블록에 적용된 SAO type을 표현하는 정보(괄호안의 숫자로 표시된 saotypeidx)가 추가적으로 전송되어야 한다. SAO는 슬라이스 단위로 적용 여부가 결정될 수 있으며, 전송한 SAO 관련 추가 정보 역시 슬라이스 단위로 전송된다.

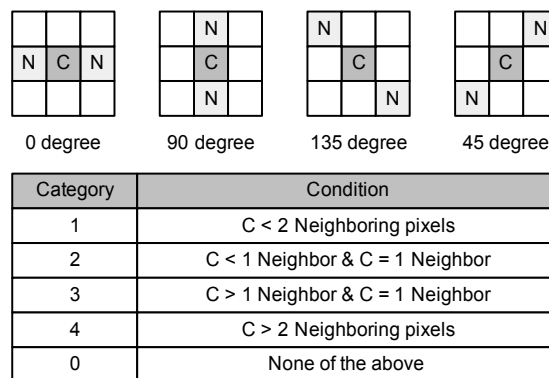
2. SAO Type

분할된 각 하위 블록에 적용 가능한 SAO type은 <그림 8>과 같다. SAO type은 SAO를 적용하지 않는 방법, EO(Edge Offset) 방법 및 BO(Band Offset) 방법으로 구분할 수 있다. EO 및 BO 방법은 해당 화소의 카테고리를 구분하기 위한 방법에 따라 다시 4 종류의 EO 방법(0-degree, 90-degree, 135-degree 및 45-degree patterns) 및 2 종류의 BO 방법(Central 및 side bands)으로 구분될 수 있다.

<그림 9>와 같이, EO는 해당 화소(C)와 인접 화소(N) 간 차이의 패턴에 따라 해당 화소가 속하는 카테고리를 분류하고, 분류된 카테고리 별로 서로 다른 offset을 적용할 수 있는 방법이다. 예를 들어, 해당 블록에 0-degree pattern의 EO가 적용되면, 해당 블록에 속한 모든 화소는 0-degree pattern에 기반해 5개의 카테고리로 분류되고, 카테고리 0을 제외한 분류된 각각의 카테고리 단위로 <식 7>과 같은 방법을 통해 각각의 offset이 결정된다. 따라서 EO가 적용되면 카테고리 0을 제외한 4개의 카테고리를 위해 4개의 offset이 결정되고 전송되어야 한다.

saotypeidx	SAO type to be used		Number of categories
0	None		0
1	Edge offset	1-D 0-degree pattern edge offset	4
2		1-D 90-degree pattern edge offset	4
3		1-D 135-degree pattern edge offset	4
4		1-D 45-degree pattern edge offset	4
5	Band offset	Central bands band offset	16
6		Side bands band offset	16

<그림 8> SAO type 정보

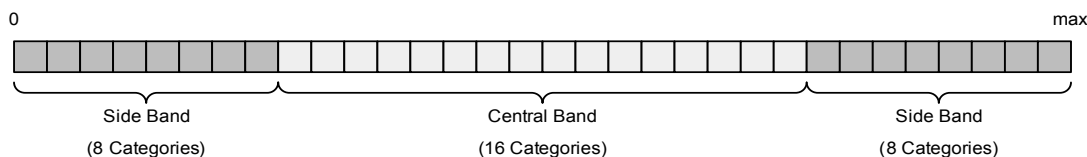


<그림 9> EO(Edge Offset)의 카테고리

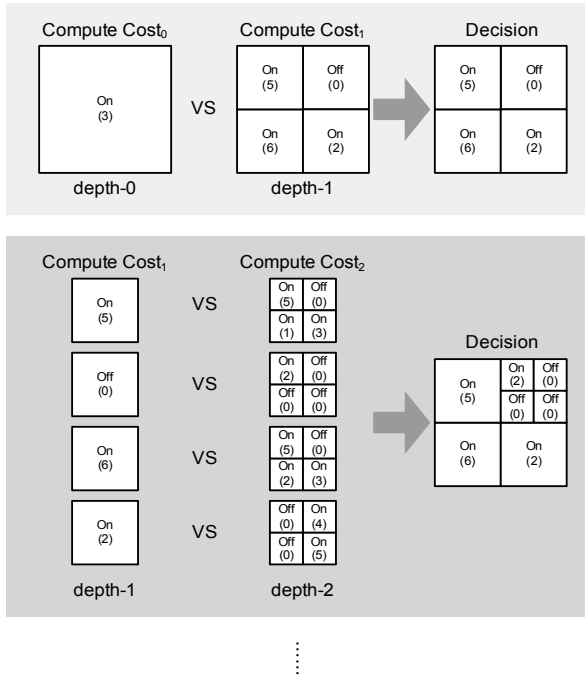
BO는 해당 화소값에 따라 해당 화소가 속하는 카테고리를 분류하고, 분류된 카테고리 별로 서로 다른 offset을 적용할 수 있는 방법이다. 즉, <그림 10>과 같이, 화소가 속할 수 있는 값을 일정한 크기의 32개의 구간(카테고리)으로 구분하고, 각 구간 단위로 <식 7>과 같은 방법을 통해 각각의 offset이 결정된다. <그림 10>과 같이, BO는 16개의 카테고리를 갖는 2개의 SAO type으로 구분될 수 있으므로, BO가 적용되면 16개의 카테고리를 위해 16개의 offset이 결정되고 전송되어야 한다.

3. SAO Block Partition 및 Type Decision

전송한 바와 같이, 부호화기는 울-왜곡 최적화 방법을 사용하여 부호화 효율 측면에서 최적인 SAO 블록분할 여부 및 SAO type을 결정할 수 있다. <그림 11>에서 보는 바와 같이, top-down 방식으로 울-왜곡 최적화 방법이 적용된다. 우선 depth=0 블록에 대해 울-왜곡 측면에서 최적인 SAO type을 결정한다. 즉, 해당 블록에 대해 <그림 8>과 같이 7개의 SAO



<그림 10> BO(Band Offset)의 카테고리



〈그림 11〉 SAO 블록분할 여부 및 SAO type 결정 과정

type이 울-왜곡 측면에서 비교되고, 가장 우수한 울-왜곡 성능을 보이는 하나의 SAO type이 해당 블록의 SAO type으로 결정된다. 이후, quadtree 기반으로 해당 블록을 네 개의 하위 블록으로 분할하고, 각각의 depth-1 블록에 대해 동일한 방법으로 울-왜곡 측면에서 최적인 SAO type을 결정한다. 마지막으로 depth-0의 울-왜곡 비용(Cost₀)과 depth-1에 대한 울-왜곡 비용(Cost₁)을 비교하여, 보다 우수한 울-왜곡 성능을 보이는 SAO 블록분할 구조 및 SAO type을 결정한다. 만약 블록이 분할되면, 〈그림 11〉과 같이, 분할된 블록을 다시 네 개의 하위 블록으로 분할하고, 각각의 depth-2 블록에 대해 울-왜곡 측면에서 최적인 SAO type을 결정한다. 이후, 각각의 depth-1 블록과 각각의 depth-2 블록의 울-왜곡 비용을 비교함으로써, 보다 우수한 울-왜곡 성능을 보이는 SAO 블록분할 구조 및 SAO type을 결정한다. 만약 블록이 분할되지 않으면, 해당 depth의 블록분할 구조가 선택되고, 더 이상 하위 블록으로 분할하지 않는다. 현재 HM3.0에서는 이러한 울-왜곡 최적화 과정을 최대 4-depth까지 반복적으로 수행할 수 있으며, 최종적으로 〈그림 7〉의 예와 같이 분할될 수 있다.

4. 관련 표준화 동향

지난 2011년 4월 스위스 Geneva에서 개최된 제5회 JCT-VC 표준화 회의에서 SAO 기술이 HM3.0에 채택됨에 따라, SAO 관련 CE-13은 종료된 상태이다.

IV. ALF(Adaptive Loop Filter) 기술

ALF는 과거부터 지속적인 연구가 이루어졌던 분야로써, ITU-T Q6/WP3/SG16 VCEG 회의에서 이미 관련된 기고서가 제안되었고 차세대 비디오 부호화 기술 후보인 KTA(Key Technology Area) 소프트웨어에 채택된 바 있다^[11~13]. VCEG 표준화 기구에 투고된 ALF 관련 기술은 최초 BALF(Block-based ALF)라는 기술의 이름이었으며, 영상을 블록 단위로 나누어 각각의 블록에 대해서 ALF를 적용 여부에 대한 정보 및 ALF에 대한 필터 계수를 부/복호화하는 기술이다. 상기 ALF에 대한 모든 정보는 slice header에 포함시켜 비트스트림에 기록한다. 원본 영상과 복원 영상과의 오차를 최소화 만드는 필터계수를 유도하여 적응적으로 블록마다 적용여부를 시그널링 하기 때문에 울-왜곡 관점에서 상당히 효율적인 압축 성능을 보여주지만, BALF 기술은 모든 블록에 대해서 ALF 적용 여부에 대한 정보를 항상 부/복호화 함으로써 부호화 효율을 감소시키는 문제가 있었다. 이 문제를 개선하기 위하여 영상을 계층적으로 블록 단위로 분할하여, 이에 대한 ALF 적용 여부 정보를 계층적으로 부/복호화를 수행하는 QALF(Quadtree-based ALF)가 제안되었다^[14]. QALF 기술은 상위 계층에 있는 블록부터 하위 계층에 있는 블록까지 순차적으로 ALF 적용 여부에 대해서 부/복호화를 수행한다. 상기 기술은 상위계층에서 ALF를 적용하지 않는다는 시그널링을 수행 할 경우, 하위 계층에 있는 블록에 대해서는 ALF 적용 여부 정보를 부/복호화할 필요가 없다. 결국, BALF에 비해서 시그널링을 해야 하는 정보의 개수가 대폭 감소하기 때문에 부호화 효율이 더욱 증가하게 된다. 현재 표준화를 진행 중인 HEVC 표준의 HM3.0에 포함된 ALF 필터링 기술은 전술한 QALF의 개념을 그대로 채용하는 동시에, 복잡한 연산 과정을 간략화시킨 방법으로서 Qualcomm 제안한 방법이 TMuC(Test Model under Consideration)^[5]에 포함된 이후 현재 HM3.0까지 유지되고 있다.

HM3.0에 있는 Adaptive Loop Filter(ALF)는 Wiener filter를 기반으로 동작하는 필터링 방법으로써, 원본 영상과 복원 영상과의 오차를 최소화시키는 최적의 계수를 〈식 9〉와 같이 계산하여 최적 필터링을 수행하고, 아울러 산출된 필터계수를 부호화 하여 전송한다. 복호화기는 전송 받은 필터 계수를 사용하여 동일하게 ALF 과정을 수행한다.

$$c_i = \operatorname{argmin} E \left[\left(x - \sum_{i=0}^{I-1} (y_i c_i) \right)^2 \right] \quad \langle \text{식 9} \rangle$$

〈식 9〉에서 x는 원본 영상 화소값을 의미하고, c_i는 i번째

필터계수를 의미하고, l 는 필터계수의 개수를, 그리고, y 는 복원 영상 화소값을 의미한다. <식 10>과 같이 각각의 필터 계수(c_0, \dots, c_{l-1})에 대해서 미분을 수행한 값이 0이 되어야만 원본 영상과 복원 영상과의 오차가 최소라고 가정할 수 있다.

$$\frac{\partial}{\partial c_k} E \left[\left(x - \sum_{i=0}^{l-1} (y_i c_i) \right)^2 \right] = 0, k \in \{0, \dots, l-1\} \quad \langle \text{식 10} \rangle$$

<식 10>의 미분 수식을 정리하면, <식 11>과 같이 도출된다.

$$E \left[\sum_{i=0}^{l-1} c_i y_i y_k \right] = E[xy_k], k \in \{0, \dots, l-1\} \quad \langle \text{식 11} \rangle$$

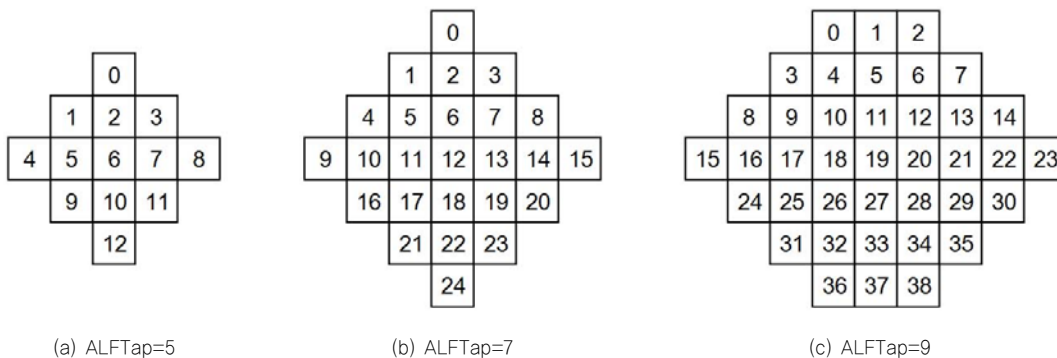
즉, 각각의 필터 계수 (c_0, \dots, c_{l-1})에 대해서 미분을 수행한 <식 11>은 <식 12>와 같이 계수들의 1차 연립방정식으로 표현될 수 있다.

$$\begin{bmatrix} E[y_0 y_0] & E[y_0 y_1] & \dots & E[y_0 y_{l-1}] \\ E[y_1 y_0] & E[y_1 y_1] & \dots & E[y_1 y_{l-1}] \\ \dots & \dots & \dots & \dots \\ E[y_{l-1} y_0] & E[y_{l-1} y_1] & \dots & E[y_{l-1} y_{l-1}] \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_{l-1} \end{bmatrix} = \begin{bmatrix} E[xy_0] \\ E[xy_1] \\ \dots \\ E[xy_{l-1}] \end{bmatrix} \quad \langle \text{식 12} \rangle$$

결과적으로, 간단하게 <식 10>의 근이 되는 필터 계수 값을 도출할 수 있게 되고, 복원 영상 화소에 상기 필터 계수를 적용함으로써 부호화 과정에서 손실되는 정보를 효과적으로 감소시킬 수 있다.

1. ALF Control Map

ALF control map은 ALF 적용 여부에 대한 on-off 신호들을 포함하고 있는 정보이다. Luma 신호의 경우, ALF 적용 여부에 대한 정보는 CTB(Coding Tree Block)에 동기화하여



<그림 12> ALF filter shape

<표 1> Chroma의 ALF 적용 여부 시그널링

symbol	정의
0	ALF를 적용 안함
1	Cb에만 ALF를 적용
2	Cr에만 ALF를 적용
3	Cb와 Cr에 ALF를 적용

CU 단위로 부/복호화를 수행한다. 단, ALF의 깊이 정보는 부/복호화를 수행함으로써, CTB의 깊이 정보와는 다르게 설정할 수 있다. ALF의 깊이 정보는 울-왜곡 관점에서 최적의 깊이를 선택하여 부/복호화를 수행한다. ALF 적용 여부를 결정하기 위해서는 ALF를 적용했을 때의 왜곡 수치와 ALF를 적용하지 않았을 때의 왜곡 수치를 산출 및 비교하여 ALF 적용 여부를 결정하고 부호화를 수행한다. Chroma 신호의 경우, ALF 적용 여부에 대한 정보는 slice 단위로 시그널링을 수행하여 부/복호화를 수행한다. 부호화기에서 ALF 적용 여부를 결정하기 위해서는 picture단위로 왜곡 수치를 산출 및 비교하여 ALF 적용 여부를 결정하고 부호화를 수행한다. ALF 적용 여부에 대한 정보는 <표 1>과 같이 정의될 수 있다. 상기 정보는 Cb 또는 Cr 정보 중 하나만 선택해서 수행할 수 있고, 또는 Cb 및 Cr 정보 둘 다 선택해서 수행할 수 있다. 또한, 필터의 종류와 필터 계수에 대해서 부/복호화를 수행한다.

2. Filter shape

Filter shape은 filtering을 적용하기 위해 필요한 참조 화소들을 어떠한 집합으로 구성할 것인지를 나타내는 정보이다. Luma의 경우, 필터의 종류는 <그림 12>와 같이 3개(5x5, 7x7 and 9x9)의 2D 다이아몬드 필터 모양 중 울-왜곡 관점에서 최적인 하나를 선택하며, 그 선택한 최적의 필터 정보를 slice 단위로 전송한다. Chroma의 경우, 필터 종류는 5x5를 가지는 사각형 모양의 필터를 사용하여 부-복호화를 수행한다.

3. Filter adaptation

ALF Filter adaptation은 filter shape들에 대해서 어떤 종류의 filter coefficient들을 사용할지 나타내는 정보이다. Luma의 경우, 필터 적응성 모드는 Block-based 적응성 모드와, Region-based 적응성 모드가 있다. Block-based 적응성 모드는 영상을 4x4 블록으로 분할 후, 각각의 블록에 해당하는 복원 화소를 사용하여 <식 13>과 같이 Activity Metric을 계산하여, 그 결과값에 따라 분류를 수행하고, 각각의 분류된 카테고리에 대해서 필터 계수를 계산 및 산출 한다.

$$\text{ActivityMetric} = L + 5 \times D \quad \langle \text{식 13} \rangle$$

L은 vertical activity(VA)와 horizontal activity(HA)가 합산된 값을 의미하고, D는 vertical activity와 horizontal activity 중 비중이 큰 것을 나타내는 값을 의미한다. 상기 vertical activity와 horizontal activity는 <식 14>로 정의된다.

$$\begin{aligned} \text{VA} &= \left| 2R_{(i,j)} - R_{(i-1,j)} - R_{(i+1,j)} \right| \\ \text{HA} &= \left| 2R_{(i,j)} - R_{(i,j-1)} - R_{(i,j+1)} \right| \end{aligned} \quad \langle \text{식 14} \rangle$$

여기서, R은 복원화소를 의미한다. Region-based 적응성 모드는 영상을 4x4로 블록으로 분할 후, <식 15>의 결과값에 따라 region을 분류한다.

$$\text{region} = \text{RTab} \left[\left(\text{id}x_y, \ll 2 \right) + \text{id}x_x \right] \quad \langle \text{식 15} \rangle$$

RTab은 0부터 15까지의 값이 할당되어 있는 배열을 의미하고, idxy와 idxx는 CU의 최대 크기 및 영상의 해상도, 그리고 현재 블록의 위치 정보를 참조하는 변수를 의미 한다. <식 15>에 의해 분류된 region에 대해서 각각 필터 계수를 계산 및 산출 한다. 상기 두가지 적응성 모드 중 울-왜곡 관점에서 하나를 선택하여 slice 단위로 부-복호화를 수행한다. Chroma의 경우, 필터 적응성 모드를 사용하지 않는다.

4. Encoding filter coefficients

ALF Filter 계수들은 exp-Golomb 코드를 사용하여 엔트 로피 부호화 된다. Luma의 경우, 첫 번째 필터에 대한 계수는 바로 해당 코드로 변환하여 부호화를 수행하지만, 그 이외의 필터에 대한 계수는 정해진 예측 필터 계수와의 차분값을 해당 코드로 변환하여 부호화 수행할지, 또는 첫 번째 필터의

부호화 방법과 동일한 방법으로 부호화를 수행할지를 시그널링하는 방식으로, 필터링 계수 부호화에 소요되는 데이터량을 줄인다. Chroma의 경우, 모든 필터에 대한 계수에 대해서 바로 해당 코드로 변환하여 부호화를 수행한다.

5. 관련 표준화 동향

현재 ALF와 관련된 CE(Core Experiments)에서는 Toshiba, Qualcomm, Panasonic 등에서 ALF control map, Filter shape, Filter adaptation, filter coefficients 부호화 방법 등의 개선을 통해 HM2.0과 동일한 부호화 효율에서 평균 3% 이상의 복잡도 감소가 가능한 기술이 제안되고 있다.

V. 결론

본 고에서는 HM3.0에 포함된 인루프 필터링 기술에 대해 살펴보았다. 디블록킹 필터링만을 수행하던 H.264/AVC 표준의 인루프 필터링 기술과 비교해 볼 때, 디블록킹 필터링 뿐만 아니라, SAO 및 ALF를 함께 포함하는 HM3.0의 인루프 필터링 기술은 보다 능동적으로 왜곡을 감소시킴으로써 화질 및 부호화 효율을 크게 향상시킬 수 있는 구조로 진화하였다. 그러나, 현재까지는 인루프 필터링 내에 포함된 각 기술간의 상호작용에 대한 분석이 미비한 상태이므로, 상호작용을 극대화할 수 있는 기술에 대한 개발이 필요할 것으로 예상된다. 특히, 현재의 인루프 필터링 기술은 매우 높은 복잡도를 요구하기 때문에, 향후의 표준화 과정에서는 성능을 유지한 채 복잡도를 감소시키는 노력이 계속되어야 할 것으로 예상된다.

참고문헌

- [1] JVT of ISO/IEC MPEG and ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification," *ITU-T Rec. H.264, ISO/IEC 14496-10 AVC*, (2003).
- [2] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "Meeting report of the first meeting of the JCT-VC," *JCT-VC doc. JCTVC-A200*, (April, 2010).
- [3] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "WD3: Working Draft 3 of High-Efficiency Video Coding," *JCT-VC doc. JCTVC-E603*, (March, 2011).
- [4] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive Deblocking Filter," *IEEE Trans. Circuits Syst. Video Tech.*, Vol.13, No.7, pp.614-619, (July, 2003).
- [5] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "Test

- Mode under Consideration," *JCT-VC doc. JCTVC-B205*, (July, 2010).
- [6] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "Samsung's Response to the Call for Proposals on Video Compression Technology," *JCT-VC doc. JCTVC-A124*, (Apr., 2010).
- [7] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "In-Loop Adaptive Restoration," *JCT-VC doc. JCTVC-B077*, (July, 2010).
- [8] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "TE10 Subtest3: Quadtree-based Adaptive Offset," *JCT-VC doc. JCTVC-C147*, (Oct., 2010).
- [9] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "CE8 Subset3: Picture Quadtree Adaptive Offset," *JCT-VC doc. JCTVC-D122*, (Jan., 2010).
- [10] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "CE13: Sample Adaptive Offset with LCU Independent Decoding," *JCT-VC doc. JCTVC-E049*, (Jan., 2010).
- [11] K. K. Pang and T. K. Tan, "Optimum loop filter in hybrid coders," *IEEE Trans. Circuits Syst. Video Tech.*, Vol.4, No.2, pp.158-167, (Apr., 1994).
- [12] T. Chujoh, A. Tanizawa and T. Yamakage, "Adaptive Loop Filter for Improving Coding Efficiency," *ITU-T SG16 contribution, C402*, (April, 2008).
- [13] Yi-Jen Chiu and L. Xu, "Adaptive Wiener Filter for Video Compression," *ITU-T SG16 contribution, C437*, (April, 2008).
- [14] T. Chujoh, N. Wada, T. Watanabe, G. Yasuda and T. Yamakage, "Specification and Experimental Results of Quadtree-based Adaptive Loop Filter," *ITU-T SG16 VCEG doc. VCEG-AK22*, (April, 2009).



양 정 업

2004년 성균관대학교 정보통신공학부 학사.
2006년 성균관대학교 전자전기공학과 석사.
2006년~현재 성균관대학교 전자전기공학과 박사과정.
<관심분야> 멀티미디어 영상압축



원 광 현

2007년 광운대학교 컴퓨터공학부 학사.
2009년 성균관대학교 휴대폰학과 석사.
2009년~현재 성균관대학교 휴대폰학과 박사과정.
<관심분야> 멀티미디어 영상압축



전 병 우

1985년 서울대학교 전자공학과 학사.
1987년 서울대학교 전자공학과 석사.
1992년 Purdue Univ. School of Elec. 공학박사.
1993년~1997년 삼성전자 신호처리연구소 선임/수석연구원.
1997년~현재 성균관대학교 정보통신공학부 교수.
<관심분야> 멀티미디어 영상압축, 영상인식, 신호처리